

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ - ПРОЦЕССОВ УПРАВЛЕНИЯ  
КАФЕДРА КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ И МНОГОПРОЦЕССОРНЫХ  
СИСТЕМ

**Берлина Анастасия Владимировна**

**Выпускная квалификационная работа бакалавра**  
**Применение технологий распределенных реестров для отслеживания**  
**истории происхождения и владения активами**

Направление 02.03.02

«Фундаментальная информатика и информационные технологии»

Образовательная программа СВ.5003.2015 «Программирование и  
информационные технологии»

Научный руководитель,  
PhD,  
доцент  
Корхов В. В.

Санкт-Петербург

2019

Введение.....	3
Постановка задачи.....	5
Глава 1. Существующие подходы к отслеживанию истории происхождения объектов.....	6
1.1. RFID.....	6
1.2. Открытые реестры и базы данных. Сервис “Автотека”.....	7
1.3. Существующие решения на основе технологий распределенных реестров.....	8
1.3.1. Everledger Diamond .....	9
1.3.2. Fishcoin Ecosystem .....	11
Глава 2. Архитектура системы.....	14
2.1. Введение.....	14
2.2 Обзор компонентов.....	15
2.2.1. Corda R3 .....	15
2.2.2. Apache Kafka.....	17
2.3. Веб - приложение .....	20
2.4. Архитектура приложения и возможные кейсы использования .....	20
2.4.1 Отслеживание истории конкретного контейнера или объекта. ....	22
2.4.2. Статистика по конкретному перевозчику .....	23
3.1. Производительность .....	25
3.2. Универсальность .....	25
3.3. Сравнение с другими решениями.....	25
3.4 Выявленные недостатки и возможное развитие.....	26
Заключение .....	27
Список литературы .....	28

## Введение

В современном мире, взаимосвязанном и объединенном глобальной сетью, основная часть экономической деятельности человечества осуществляется посредством коммерческих сетей, стирающих географические, национальные и юридические границы. Такие сети чаще всего переплетаются на торговых площадках, где участники рынка обмениваются принадлежащими им ресурсами, чаще называемыми активами, а также заявляют свои права на них. Для участников таких систем основополагающим является свойство прозрачности: важно понимать, что актив не пропадет, никто из участников не окажется мошенником и не сможет подменить товар. В то же время и конечные потребители хотят знать о приобретенном товаре больше. Возьмем, к примеру, рыбную промышленность. Покупатель хочет быть уверен в том, что рыба, которую он купил, не была выловлена в закрытых водах или для её поимки не использовались незаконные способы добычи.

Отслеживание довольно затруднено, когда информация о материалах неполна или отсутствует; более того, возможности отслеживания и прозрачности ограничены сложностью в сети поставок. Трудно идентифицировать всех поставщиков и покупателей и отслеживать их действия, так как они могут находиться в совершенно разных концах планеты, и потому такие цепочки могут быть сложными, многоуровневыми, и, более того, глобальными. К примеру, отследить единственный источник - производителя кофейных зерен - менее сложно, чем многонациональный конгломерат, который объединяет бобы от нескольких производителей из нескольких стран. Сложность сетей цепочки поставок, состоящих из различных участников (т. е. поставщиков, дистрибьюторов, производителей и конечных потребителей) заключается в том, что эти сети состоят из скрытых элементов, из-за чего возникают определенные риски для всех участников цепи, ведь для надежности обмена активами нужно установить

достоверное происхождение и исключить возможность мошенничества и подделки.

В данной работе будут рассмотрены уже существующие решения для отслеживания истории происхождения и владения активами и предложен альтернативный вариант на базе технологий распределенных реестров, призванный сделать систему максимально прозрачной и устойчивой к подмене данных.

## Постановка задачи

Вышеуказанные проблемы можно решить использованием набирающих сейчас популярность технологий распределенных реестров. Данные технологии предоставляют возможность фиксировать каждое действие (обмен активами, передача прав владения другому лицу) в реестре в виде транзакций. Каждая транзакция сначала проверяется участниками на валидность, и в случае успеха, записывается в реестр. Каждый участник сети хранит актуальную версию реестра на данный момент. Данные функции решают проблему доверия между участниками сети, а также проблема фальсификации информации об активах.

Именно поэтому целью данной работы является разработка универсальной системы для отслеживания истории происхождения и владения активами на основе технологий распределенных реестров. В данной системе должен быть представлен следующий функционал:

- Отслеживание истории владения и передачи прав на конкретный актив
- Возможность отследить, на каком этапе и что происходило, и выделить “виновного” в случае, если по итогу объект оказался испорчен
- Сбор статистики по конкретному перевозчику/элементу транспортной сети

Поэтому в задачи работы входит:

- Анализ существующих решений для отслеживания истории активов, выявление их преимуществ и недостатков
- Выбор компонент, наиболее подходящих для реализации подхода
- Разработка приложения, отвечающего поставленным требованиям
- Анализ разработанного решения, сравнение с существующими

# **Глава 1. Существующие подходы к отслеживанию истории происхождения объектов.**

## **1.1. RFID**

**RFID (радиочастотная идентификация)-метка** - миниатюрное запоминающее устройство. Она состоит из микрочипа, который хранит информацию, и антенны, с помощью которой метка передает и получает данные. Эти компоненты позволяют тегу реагировать и излучать радиосигналы. В памяти RFID-метки хранится уникальный номер и информация. Когда метка попадает в зону регистрации, эта информация принимается RFID-считывателем. [1]

RFID-метки в настоящее время широко используются для отслеживания цепи поставок в грузовых перевозках. Товар отслеживается RFID-системами на каждом из этапов: на производстве, при перевозке, в момент складской обработки и в момент продажи. Полученная информация сохраняется в информационной системе текущего звена цепи распределения. К отдельным отчетам, строящимся на основе данной информации, открывается доступ для авторизованных пользователей, в том числе и внешних (для предприятий-партнеров).

Существует возможность совмещения RFID-технологий с GPS. В таком случае кроме уже перечисленных эффектов достигается возможность круглосуточного контроля над транспортным средством с определением не только точного местоположения в каждый конкретный момент времени, но и полного списка параметров транспортного средства и его груза. [2]

### **Достоинства:**

- **Возможность скрытого размещения не извлекаемого идентификатора**, в том числе допускается его встраивание в объект (например, внутрь пластмассовой детали при ее отливке).
- **Высокая степень безопасности.** Уникальное неизменяемое число-идентификатор, присваиваемое метке при производстве, гарантирует

высокую степень защиты меток от подделки. Также данные на метке могут быть зашифрованы.

- **Возможность отслеживания объекта в реальном времени.**

#### **Недостатки:**

- **Подверженность помехам в виде электромагнитных полей.**  
Системы радиочастотной идентификации могут быть чувствительны к помехам в виде электромагнитных полей от включенных компьютеров.
- **Возможность подмены груза внутри контейнера, на котором находится метка.** Метки, уведомляющие о незаконном вскрытии, только вводятся в эксплуатацию.
- **Высокая стоимость.** Использование RFID-меток целесообразно для дорогих, уникальных товаров или больших объемов перевозок, но не для розничных товаров.
- **Возможность манипулирования данными в системе.**
- **Недоступность информации для конечного потребителя.**
- **Отсутствие единого стандарта.**

## **1.2. Открытые реестры и базы данных. Сервис “Автотека”.**

Информацию об активах, права на которые закрепляются при помощи договоров, утверждаемых уполномоченными лицами, можно найти в открытых реестрах, например, в реестре залогов или на сайте судебных приставов РФ.

В таких открытых базах данных фиксируется, от кого и к кому перешел определенный актив, сохраняются записи о действиях производимых с ним и т.д.

К примеру, в настоящее время в нашей стране набирает популярность сервис “Автотека”, который позволяет по государственному номеру автомобиля выдать наиболее полный отчет о его истории. Данное

приложение собирает информацию со всех доступных открытых ресурсов и базах данных, открытых сервису его партнерами. В частности, можно узнать реальное количество владельцев автомобиля, его общий пробег, а также информацию о том, находился ли он под арестом, участвовал ли в ДТП, использовался ли в качестве такси и многое другое. [3]

В настоящее время к сервису присоединилось множество независимых станций ТО, дилерских центров, страховых компаний и некоторых государственных организаций.

#### **Достоинства:**

- **Удобство использования.** Конечному потребителю не приходится тратить время на посещение всевозможных открытых ресурсов, всё собрано в одном месте.
- **Доступ к закрытой информации компаний-партнеров.** “Автотека” использует недоступные для потребителей базы данных.

#### **Недостатки:**

- **Не обновляемая база автомобилей.** База данных сервиса не обновлялась с 2017 года.
- **Недействительная информация.** В предлагаемых отчетах информация может быть непроверенной и неточной. Ложная информация может негативно сказаться на конечном выборе покупателя.
- **Нестабильная работа сервиса.** “Автотека” часто работает медленно, к тому же одни и те же сведения об автомобилях могут дублироваться несколько раз.
- **За отчет о каждом автомобиле нужно вносить определенную плату.**

### **1.3. Существующие решения на основе технологий распределенных реестров**

Distributed ledger technology (DLT) — это протокол для построения реплицированной и совместно используемой системы учета записей. Такая



система может быть использована для записи широкого спектра объектов, таких как владение активами, транзакции передачи активов и контракты соглашения.

DLT - база данных, которая распределена между несколькими сетевыми узлами или вычислительными устройствами. Каждый узел получает данные из других узлов и хранит полную копию реестра. Обновления узлов происходят независимо друг от друга. Все транзакции проверяются сетевыми узлами и записывается в распределенную базу данных.[4]

Ключевая особенность распределенного реестра — отсутствие единого центра управления. Каждый узел составляет и записывает обновления реестра независимо от других узлов. Затем узлы голосуют за обновления, чтобы удостовериться, что большинство узлов согласно с окончательным вариантом. Голосование и достижение согласия в отношении одной из копий реестра называется консенсусом, этот процесс выполняется автоматически с помощью соответствующего алгоритма. Как только консенсус достигнут, распределенный реестр обновляется, и последняя согласованная версия реестра сохраняется в каждом узле.[5]

### **1.3.1. Everledger Diamond**

Бриллианты - драгоценные камни, отличающиеся своей красотой и твердостью, которые могут классифицироваться по весу, форме, четкости и цвету. Они используются не только в ювелирном производстве, но и в промышленности. В настоящее время на мировом алмазном рынке широко распространяются синтетические бриллианты и бриллианты, добытые в зонах военных действий (так называемые кровавые алмазы). Чтобы удостовериться в подлинности камня, необходимо знать его географическое происхождение; его источник, то есть переработанный он, добытый или синтетический, и условия добычи. Все эти факторы можно отнести к такому понятию, как происхождение камня, к которому также можно отнести и

записи о правах владения. Происхождение обеспечивает условия для отслеживания, которое детализирует историю кусочка камня со дня его добычи до настоящего времени.

Для этого были разработаны ключевые правила и отраслевые стандарты. В частности, Кимберлийский процесс (КП), который в своих руководящих принципах производства и торговли гарантирует, что алмазное сырье из зон конфликта не попадет в руки потребителей. По процессу Кимберли требуется, чтобы каждый экспорт алмазов сопровождался сертификатом Кимберлийского процесса (KPCS). Однако, в связи с тем, что сертификация занимает длительное время (от 2-3 недель до 3 месяцев и более), довольно распространены подделки бумажных сертификатов.[6]

Everledger предложил свой протокол для упрощения отслеживания бриллиантов - Diamond-Time-Lapse-Protocol (DLTP), созданный на основе технологии блокчейн. Данное решение создает собственный реестр для того, чтобы следить за передвижением алмазов с определенными метаданными. В эти метаданные может входить информация о происхождении и источнике добычи; о таких атрибутах драгоценных камней, как карат и огранка; данные о таких процессах, как резка и полировка; ремесленная работа и сертификация. Эта информация затем используется для создания уникальной записи в публичных и приватных реестрах.[7]

Первым этапом в производстве алмазов является разведка, где добывать сырье и как его классифицировать (на основании размера, цвета, прозрачности и т. д.). KPCS выдает сертификат на каждый добытый алмаз. Решение Everledger позволяет собирать метаданные с бриллиантов и присваивать каждому из них уникальный идентификационный номер для последующего сохранения в блокчейне. Бриллианты помечены так, чтобы была затем возможность связать их с цифровыми записями.

Следующий этап - резка и полировка. Каждой новой части разреза присваивается дополнительный уникальный идентификационный номер. После этого кусочки алмаза идут на полировку или огранку. Затем

бриллианты проходят через процесс контроля качества, который подлежит валидации.

Бриллианты, не прошедшие процедуру проверки, возвращаются на предыдущий этап, в то время как те, которые проходят процесс проверки, получают итоговую оценку. Алмазы сертифицируются Геммологическим институтом Америки (GIA) по результатам научных анализов, таких как спектроскопия, морфология и структура роста кристаллов. Сертификат загружается в блокчейн Everledger.[8]

#### **Достоинства:**

- **Решение охватывает все этапы разработки:** от добычи до поступления на рынок.
- **Невозможно подделать информацию в системе ни на одном из этапов.**
- **Все записи о приобретенных бриллиантах доступны конечному потребителю.**

#### **Недостатки:**

- **Решение узко специализировано:** многое завязано на специфических физических характеристиках бриллиантов.

### **1.3.2. Fishcoin Ecosystem**

Индустрия морепродуктов — это функционирующая бизнес-экосистема, в рамках которой производится сбор морепродуктов, их обработка, продажа, экспорт, импорт, и, в конечном счете, продажа потребителям. В настоящее время объем данных, перемещаемых по цепочкам поставок, ничтожен по сравнению с обилием возможных данных, которые могут быть получены и переданы в целях повышения эффективности, качества, безопасности и устойчивости в мировой индустрии морепродуктов.

Fishcoin Ecosystem была разработана для создания экосистемы данных, которая работает параллельно с бизнес-экосистемой индустрии морепродуктов, чтобы сделать ее более эффективной и более ценной.

Например, в каждой транзакции между независимыми заинтересованными сторонами параллельно происходит две вещи: реальный мировой актив (вылов морепродуктов) обменивается на фиатную валюту, и запись этого урожая становится доступной с помощью смарт-контракта для следующего участника в цепочке в обмен на токены Fishcoin. Поскольку токены Fishcoin представляют собой ценность для следующего получателя, готовность этой стороны обменять токены Fishcoin на данные указывает на то, что данные являются точными, поскольку неточность может помешать последующей передаче вместе с активом, а в некоторых случаях может даже предотвратить продажу актива в целом. [9]

Другим важным фактором является необходимость рыночного механизма для определения соответствующего вознаграждения, необходимого для стимулирования сбора и обмена данными на каждом узле сети. Например, более богатый производитель на Аляске, вероятно, потребует больше стимулов для записи данных в блокчейн, чем мелкомасштабный производитель в развивающейся стране. Кроме того, не все данные имеют одинаковую ценность. Если субъектам в экосистеме был назначен определенный уровень вознаграждения за каждое событие передачи данных (например, один токен для каждого нового размещения данных), то такой подход ограничит способность экосистемы к адаптации к рыночным условиям и возможность развиваться с течением времени. Из-за этого количество токенов, необходимых для каждого, определяется участниками, а не центральным органом. Это также помогает экосистеме самостоятельно приспосабливаться к колебаниям цен в токене с течением времени, так что сеть остается жизнеспособной. [10]

#### **Достоинства:**

- Система покрывает крупномасштабные международные цепочки поставок: от небольших фермерских хозяйств до многонациональных конгломератов

- **Разработан механизм мотивации участников системы для предоставления валидной информации**

**Недостатки:**

- **Свой токен в условиях пресыщения рынка**
- **В данный момент приложением невозможно воспользоваться: при попытке запуска либо выводится ошибка, либо идет бесконечная загрузка.**

## Глава 2. Архитектура системы

### 2.1. Введение

Для отслеживания истории происхождения активов нам в первую очередь необходимы данные о переходе прав владения на актив. Реализация части системы, генерирующей события, связанные с предметом, лежит в рамках компетенции другой работы, которая вместе с данной образуют единую систему.

За основу взят кейс грузоперевозок: есть контейнеры, которые транспортируются конкретными компаниями-перевозчиками; имеется корневой контейнер (их может быть несколько в рамках одного средства перевозки), он является основным для всех вложенных. Контейнеры могут иметь еще вложенные контейнеры или предметы, при этом каждый из них является узлом Corda. Они соединены в локальную сеть и образуют распределенный реестр.

Каждый предмет обладает сертификатом, подписанным и выданным certification authority, которое является конкретным для каждого типа предмета. Сертификат обладает возможностью валидации.

Для проверки содержимого уполномоченному лицу необходимо подтвердить свою личность и право проверки, предоставив свой сертификат, выданный ему специфическим certification authority. В случае успешной проверки сертификата узлы всех контейнеров подпишут транзакцию, и в ответ уполномоченному лицу вернутся сертификаты всех предметов, лежащих в корневом контейнере, и во всех что в него вложены. На основе этих данных он сможет сделать вывод, можно ли пропускать корневой контейнер дальше по пути его следования.

Результат проверки также записывается в реестр таким же механизмом для дальнейшего анализа.

Предположим также, что на контейнерах имеются ИОТ сенсоры, которые фиксируют нестабильные или экстраординарные ситуации в течение перевозки. Примерами таких ситуаций могут быть: сотрясение контейнера,

которое могло отразиться на перевозимых хрупких предметах , или отключение системы охлаждения объектов, которые нужно перевозить при определенной температуре.

Все эти события записываются в распределенный реестр, позволяя получателю актива узнать историю его передвижения, и в случае, если он оказался испорчен, иметь возможность предъявить свои претензии конкретной компании-перевозчику.

Также для того чтобы анализировать информацию в распределенном реестре и восстанавливать историю владения объектом, необходимо, для начала, собрать всю информацию со всех узлов сети в одном месте.

Поэтому условно можно выделить два этапа разработки системы:

1. Сбор информации в распределенном реестре
2. Анализ полученных данных и восстановление истории владения активом

## **2.2 Обзор компонентов**

Для реализации системы, реализованная в рамках этой работы, были выбраны следующие компоненты:

1. Платформа Corda на основе блокчейна Ethereum, использующая JVM смарт-контракты. Это децентрализованная база данных или распределенный реестр.
2. Apache Kafka — брокер сообщений, реализующий паттерн Producer-Consumer. Это Open Source разработка, созданная компанией LinkedIn на JVM стеке (Scala).

### **2.2.1. Corda R3**

Corda R3 (R3CEV LLC) – финансово-технологическая исследовательская компания, базирующаяся в Нью-Йорке. Она ведет консорциум из более чем 70 крупных мировых финансовых институтов в области исследований и разработки использования базы данных блокчейн в

финансовой системе. R3 была основана в 2014 году Дэвидом Э Раттером, сейчас директором является Ричард Г. Браун.

Основные понятия в модели Corda:

- *State objects* (объекты состояния), представляющие собой соглашения между двумя или более сторонами, которые регулируются смарт-контрактом.
- *Транзакции*, которые описывают переходы state object-ов от стороны к стороне на протяжении их жизненных циклов.
- *Smart-contract* определяет правила, в соответствии с которыми происходит завершение транзакции. Контракт ссылается и реализует *Legal Prose* - документа, имеющего юридическую силу и описывающего права и обязанности сторон.
- *Flow Framework*, компонент, который упрощает процесс написания сложных многошаговых протоколов между несколькими сторонами через Интернет, и позволяет сторонам координировать действия без центрального органа.[11]

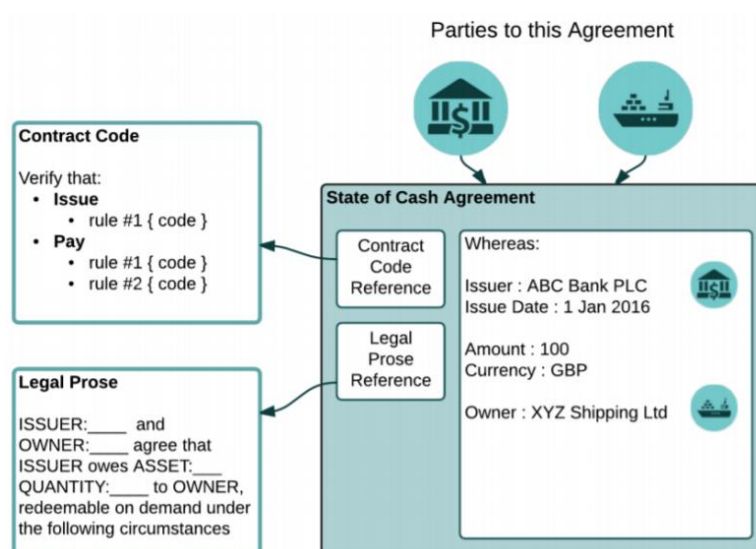


Рис.1. Схематичное представление state object, смарт-контракта и legal prose

Транзакции, которые совершаются в платформе Corda, не транслируются всем ее участникам. Записи доступны лишь тем участникам сети, которые обладают законным правом на их просмотр и управление.



Таким образом, ноды “видят” транзакции, только если они имеют к ним отношение либо если эти транзакции взаимосвязаны с деятельностью ноды.

Транзакция будет добавлена в реестр, если она:

- контрактно валидна
- подписана всеми необходимыми участниками
- не содержит двойных трат (double-spends)

Ключевой особенностью Corda является то, что платформа не использует блокчейн. Вместо этого применяются специальные нотариальные ноды. В данном случае нотариат представляет собой децентрализованный сервис, играющий ту же роль, что и майнеры в других блокчейн-системах. То есть нотариальные ноды подтверждают, что транзакция может быть принята и пропущена дальше. Также нотариат используется для создания меток времени.[12]

Основные преимущества Corda R3:

- Смарт-контракты могут быть составлены из документов, имеющих юридическую силу
- Нет собственного токена
- Активно развивающийся проект с открытым исходным кодом

### **2.2.2. Apache Kafka**

Apache Kafka – распределенный брокер сообщений, работающий по принципу «публикация-подписка» с открытым исходным кодом.

Обмен сообщениями publisher/subscriber - это шаблон, который характеризуется отправителем (издателем) фрагмента данных (сообщением), конкретно не направляя его получателю. Вместо этого издатель классифицирует сообщение каким-то образом, и этот получатель (подписчик) подписывается на получение определенных классов сообщений.

Publisher/subscriber системы часто имеют брокера, центральную точку,

которая обеспечивает механизм копирования сообщения из входящих каналов в выходные [13].

Ключевые особенности Kafka:

- Распределённая система, которую легко масштабировать.
- Поддержка высокой пропускной способности как со стороны источников, так и для систем-подписчиков.
- Поддержка объединения подписчиков в группы.

Основные компоненты архитектуры:

- Topic - Поток сообщений определенного типа, категория, в соответствии с которой публикуется то или иное сообщение.
- Producer – процесс, публикующий сообщения в тему
- Broker – один сервер из кластера Kafka Consumer – процесс, который читает сообщения из темы. Может подписаться на одну или несколько тем и использовать сообщения, забирая данные от брокеров.
- Partition – часть темы, упорядоченная, неизменная последовательность сообщений. Внутри каждого partition сообщения идентифицируются уникальным номером – offset.[14]

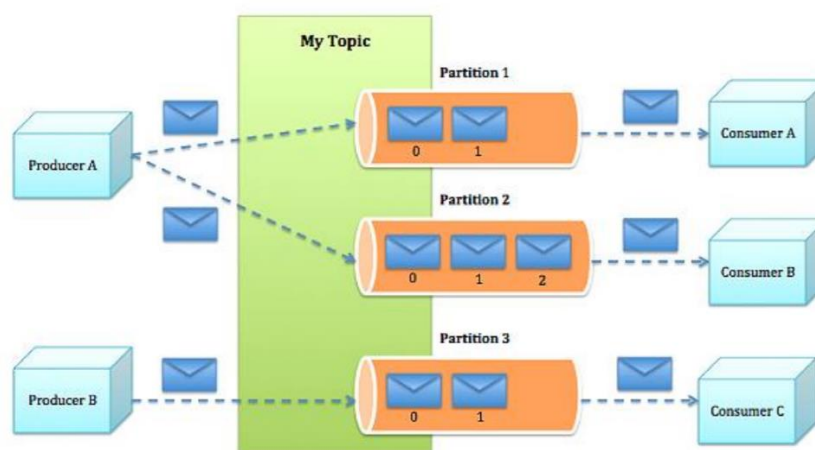


Рис. 2. Функциональная схема компонентов Apache Kafka

Единица измерения информации в Kafka - *сообщение*. Сообщение - это просто массив байтов, поэтому тип данных содержащийся в нем не имеет для платформы никакого значения. Сообщение может быть записано в определенной схеме, например, JSON или XML. Также в сообщении может

быть указана метainформация о сообщении - ключ. Ключ тоже массив байтов, который помогает записывать сообщения в *темы* определенным разработчиком образом. [15]

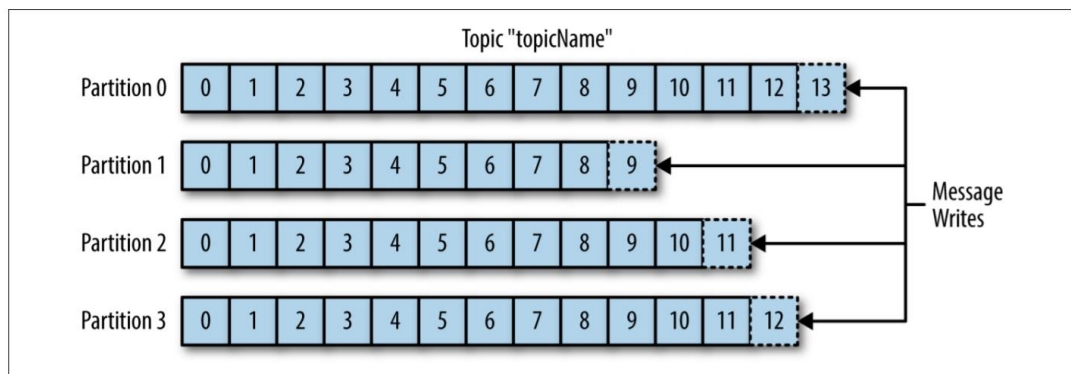


Рис. 3. Тема с несколькими разделами.

Разделы - это также способ, которым Kafka обеспечивает масштабируемость. Каждый раздел может быть размещен на отдельном сервере, что означает, что одна тема может быть масштабирована по горизонтали между несколькими серверами, чтобы обеспечить производительность, намного превышающую возможности одного сервера.

Однако для нашего приложения, недостаточно просто читать хранить и записывать потоки данных, нужно иметь возможность производить действия над этими потоками и обрабатывать их в реальном времени. Для этого в Apache Kafka существует такой механизм, как Stream Processing.

Этот компоненты платформы позволяет фильтровать, агрегировать и производить другие операции над данными в реальном времени.

Основные достоинства Kafka:

- Является распределенной системой, которую легко масштабировать
- Поддерживает быструю передачу большого числа сообщений: около 100 000 в секунду
- Как и Corda R3, Kafka разработана на стеке JVM, что существенно упрощает их интеграцию.

## 2.3. Веб - приложение

Мною было реализовано Java Spring Boot веб-приложение, которое включает в себя взаимное функционирование Corda и Kafka, обработку и анализ информации лежащей в Kafka.

Также был разработан веб-сайт через который пользователь может взаимодействовать с системой и наглядно увидеть всю историю передвижения объекта.

## 2.4. Архитектура приложения и возможные кейсы использования

Рассмотрим обозначенный нами первый этап - сбор информации со всех узлов распределенной сети.

В нашей системе аутентификации есть несколько видов State Object. Основными являются:

- put container
- change carrier
- event
- delete container

У каждого из событий существует несколько характеристик, способных указать на порядок их происхождения: временная метка, обозначающая когда оно было зарегистрировано, имя контейнера или перевозчика и гео-метка, указывающая на место, где событие произошло. Также у каждого из объектов состояния существует большое количество полей, связанных с подтверждением транзакций и их характеристиками. Однако эти данные никак не могут быть использованы для восстановления истории происхождения, поэтому будем использовать только поля объектов событий, что указаны выше.

Для каждого из типов события, в соответствии с названием, создается своя тема в Kafka, куда будут писаться события конкретно этого вида.

В каждую тему будут писаться state objects определенного типа, для того чтобы после при чтении данных из Kafka и дальнейшем анализе, мы сразу знали с каким типом объекта имеем дело, что позволит избежать лишних действий. К тому же, основным принципом Kafka является то, что в конкретном topic-е должны лежать объекты со схожей структурой и смыслом.

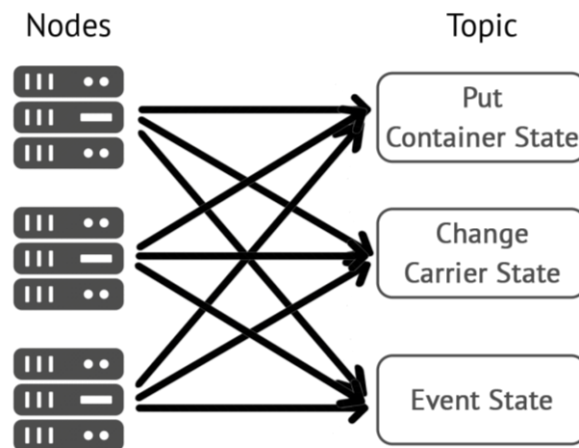


Рис. 4 Схема сбора информации с узлов

Сообщения, отправляемые в Kafka имеют вид ключ - значение. В качестве ключа обозначим id транзакции, а значением - данные, которые она в себе содержит. В данные не попадают важные или конфиденциальные данные, только те, которые нужно знать, для восстановления истории владения.

Как было упомянуто выше, в нашей системе контейнеры могут иметь вложенные в них контейнеры или предметы, и если происходит, например, смена владельца, у корневого то это будет записано и у вложенных в него. Поэтому некоторые транзакции попадающие в Kafka дублируют друг друга.

Для того, чтобы не делать лишних операций по обработке state objects, нужно сразу отфильтровать сообщения, имеющие одинаковые ключи. Для этого используется вышеупомянутый компонент Kafka: streams processing.

Также для повышения производительности отправка каждого типа события в topic осуществляется в параллельном режиме. Аналогично в

многопоточном режиме из Kafka считываются сообщения consumer-ами из каждой темы в свою очередь, специфическую для каждого типа сообщения.

События связываются по имени актива (которое является уникальным в системе), по дате и времени, и по геометкам: долготе и широте.

#### **2.4.1 Отслеживание истории конкретного контейнера или объекта.**

Для того, чтобы созданной системой было удобно пользоваться конечному потребителю, было предложено следующее решение: на каждый контейнер или объект прикреплять QR-code, просканировав который, пользователь будет перенаправлен на сайт, где он сможет увидеть историю передвижения своего актива. На данный момент, у нас нет выделенного под наше приложение хоста, поэтому QR код на рисунке ниже, приведен как пример.



Рис.5. QR-code, который ведет на сайт с историей объекта

Однако очевидно, что такой способ предоставления информации потребует дополнительных затрат: как трудовых, так и денежных, так как QR-коды нужно кому-то специально наносить, печатать и т.д. Поэтому есть и альтернативный метод получения истории передвижения через веб-сайт - интерфейс, в котором пользователю нужно вручную ввести имя объекта.

Track your order

Item

Supply chain

Name:

myContainer

Track container by name

Name	Date	Type	Geolocation
myContainer	May 15, 2019, 6:35:00 PM	Tracking started	NY USA 40.78788 -74.01431
Gruzovichkoff	May 15, 2019, 6:37:30 PM	Change carrier	NY USA 40.78788 -74.01431
OnTime	May 17, 2019, 3:22:30 PM	Change carrier	LA USA 34.052235 -118.243683
MakeMeHappy	May 21, 2019, 9:45:30 PM	Change carrier	Rio Brazil 22.90278 43.2075
myContainer	May 22, 2019, 6:35:00 PM	Tracking stopped	Rio Brazil 22.90278 43.2075

Рис.6. История для объекта под названием “myContainer”

Пользователь увидит всю историю актива: от момента, когда его передали первому перевозчику (Tracking started), события смены компаний-перевозчиков (Change carrier), и до момента, когда объект был доставлен получателю (Tracking stopped). Также, если на пути происходили события, которые могли бы повлиять на сохранность объекта - это тоже отражается в его истории. Соответственно, можно сразу увидеть какой перевозчик выполнял свои обязанности недобросовестно.

Track your order

Item

Supply chain

Name:

myContainer

Track container by name

Name	Date	Type	Geolocation
myContainer	May 15, 2019, 6:35:00 PM	Tracking started	NY USA 40.78788 -74.01431
Gruzovichkoff	May 15, 2019, 6:37:30 PM	Change carrier	NY USA 40.78788 -74.01431
Strong shaking	May 16, 2019, 6:36:21 PM	Warning!	
OnTime	May 17, 2019, 3:22:30 PM	Change carrier	LA USA 34.052235 -118.243683
MakeMeHappy	May 21, 2019, 9:45:30 PM	Change carrier	Rio Brazil 22.90278 43.2075
myContainer	May 15, 2019, 6:35:00 PM	Tracking stopped	Rio Brazil 22.90278 43.2075

Рис.7. История для объекта, в пути которого происходили события

## 2.4.2. Статистика по конкретному перевозчику

Также если пользователю системы захочется посмотреть статистику конкретного перевозчика, то он тоже сможет это сделать. В основе

составления статистики лежит сравнение участка маршрута, на котором объект транспортировался конкретным перевозчиком, со всеми остальными: если у большинства перевозчиков (больше половины) на таком же участке пути не было исключительных событий (тряски), или эти ситуации никак не зависят от условий перевозки (критические ситуации, например, отключение системы охлаждения), то эта часть маршрута помечается.

Track your order [Item](#) [Supply chain](#)

Name: OnTime

Statistics

From	To	Date	Event
NY USA 40.78788 -74.01431	LA USA 32.78788 -54.01431	May 10, 2019, 6:36:21 PM	Shaking
LA USA 32.78788 -54.01431	Rio Brazil 21.78788 -114.01431	May 14, 2019, 12:32:11 AM	
Rio Brazil 21.78788 -114.01431	Capetown Respublic of SA 55.78788 -93.01431	May 15, 2019, 6:14:51 AM	

Рис.8. Статистика по конкретному перевозчику



## **Глава 3. Анализ полученного решения**

В данной главе будет проанализирована полученная система, проведено сравнение с другими решениями, также будут выявлены ее достоинства и недостатки.

### **3.1. Производительность**

Для того чтобы данный прототип системы использовать на реальных задачах, необходимо прежде убедиться, что она будет так же хорошо работать и на большом количестве объектов в системе, что также увеличивает и число событий в их истории.

Одним из узких мест в системе является сбор информации с узлов сети и дальнейшая обработка, но так как каждый тип события обрабатывается параллельно, то время работы приложения на данных этапах будет расти линейно, что можно считать удовлетворительным.

### **3.2. Универсальность**

Одним из неоспоримых преимуществ нашей системы является возможность переноса основных ее принципов на смежные области и подобные приложения. В данном решении для восстановления истории владения активом используются наиболее общие характеристики: имя объекта, дата и время конкретного события и геоданные. Такой подход позволяет расширить область работы приложения с кейса грузоперевозок до любой другой, основанной на распределенном реестре Corda R3, и предоставляющий в своих state object указанные характеристики.

### **3.3. Сравнение с другими решениями**

При анализе существующих решения для восстановления истории владения объектом были выявлены такие основные недостатки как:

- Привязка истории к конкретным характеристикам актива. Наше решение избавлено от этого недостатка (описано в предыдущем параграфе)

- Нестабильная работа приложения. Благодаря своим возможностям масштабируемости и производительности данное приложение будет стабильно работать даже при больших объемах информации об объектах.
- Недостоверная информация об истории объекта. Данная проблема решается благодаря использованию технологий распределенных реестров, где подменить или сфальсифицировать информацию об активе не представляется возможным.

### **3.4 Выявленные недостатки и возможное развитие**

Несмотря на все выявленные достоинства, сложно отрицать что предложенное решение имеет ряд недостатков.

Одним из основных является отсутствие мотивации компаний перевозчиков для установки дополнительного оборудования для сбора информации и регистрации всех событий в нашей системе. Перевозчикам попросту невыгодно предоставлять пользователям информацию о том, насколько добросовестно они выполняют свои обязанности, потому что пользователям станет проще высказывать свои претензии.

В дальнейшем хотелось бы добавить более глубокую аналитику по элементам транспортной сети, расширить кейсы использования с грузоперевозок до отслеживания истории владения конкретными людьми, например, для нематериальных активов.

## **Заключение**

В данной работе были рассмотрены существующие подходы и готовые решения для отслеживания истории происхождения и владения активов, построенные на основе технологий распределенных реестров и без них. В результате анализа были выявлены основные недостатки: недоступность всей необходимой информации конечному пользователю, узкая направленность решений, нестабильная работа.

Были выявлены основные критерии, которым должно отвечать разрабатываемое решение. Также был осуществлен подбор компонентов, лучше всего отвечающих поставленным требованиям, и построена архитектура искомой системы.

В результате было создано веб-приложение на основе технологий распределенных реестров, позволяющее отследить историю конкретного объекта и собрать статистику по определенному элементу транспортной сети. Предложенное решение было проанализировано, были выявлены основные достоинства и недостатки, а также предложены возможные пути развития для улучшения его работы.

Поставленные задачи были выполнены, а цель работы достигнута.

## Список литературы

1. Syndicate RFID[Электрон. ресурс]/RFID tags for container tracking & logistics - URL: <https://syndicaterfid.com/RFID-tags-for-container-tracking-and-logistics-management#>( Дата обращения 18.05.2019)
2. Логист [Электрон. ресурс]// Применение RFID в цепи поставок - URL: <http://logist.ru/articles/primenenie-rfid-v-seri-postavok>(Дата обращения 10.03.2019)
3. Сервис Автотека[Электрон. ресурс] - URL: <https://autoteka.ru/about> ( Дата обращения 21.04.2019)
4. An Overview of Blockchain Technology:Architecture, Consensus, and Future Trends [Электрон. ресурс]//Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen and Huaimin Wang - URL: <https://www.researchgate.net/publication/318131748> (Дата обращения 17.05.2019)
5. David Moskowitz, Tim Swanson, R.C.: A Gentle Introduction to Blockchain Technology[Электрон. ресурс] - URL: <https://bitsonblocks.net/a-gentle-introduction-to-blockchain-technology/> (Дата обращения 19.04.2019)
6. What is the Kimberley Process? [Электрон. ресурс] - URL: <https://www.kimberleyprocess.com/en/what-kp/> / (Дата обращения 11.04.2019)
7. A. O. Orege, F. Milani: Blockchain-based Provenance Solution for Handcrafted Jewellery: master thesis - 78с.
8. Everledger Diamond [Электрон. ресурс] - URL: <https://diamonds.everledger.io/#technology> / (Дата обращения 19.04.2019)
9. FishCoin Ecosystem [Электрон. ресурс]/ Whitepaper - URL: <https://fishcoin.co/files/fishcoin.pdf> (Дата обращения 19.04.2019)
10. Fishcoin Token Velocity [Электрон. ресурс] - URL: <https://medium.com/fishcoin/fishcoin-token-velocity-c607e4051f33> (Дата обращения 19.04.2019)
11. R. G. Brown: The Corda Platform: An Introduction/ R. G. Brown - 21 с.

12. Официальный сайт Corda R3 [Электрон. ресурс] - URL:  
<https://docs.corda.net/> (Дата обращения: 18.03.2019)
13. Publisher-Subscriber pattern[Электрон. ресурс] - URL:  
<https://docs.microsoft.com/en-us/azure/architecture/patterns/publisher-subscriber>  
(Дата обращения: 02.04.2019)
14. N. Narkhede Kafka: The Definitive Guide // N. Narkhede, G. Shapira, T. Palino - 322 с.
15. Официальный сайт Apache Kafka [Электрон. ресурс] - URL:  
<http://kafka.apache.org/> (Дата обращения: 18.03.2019)